

Claims

1. A network interface card comprising:

an upper layer protocol (ULP) handler;

a TCP handler capable of interfacing with said ULP handler; and,

a link handler,

wherein the ULP handler, the network interface card is adapted to take over and perform at least one session layer function of a host computer connected to a network.

2. The network interface card of claim 1, wherein the network interface card is a layer 5 network interface card in a network implementing an open systems interconnection (OSI) protocol.

3. The network interface card of claim 1, wherein data from the network is received and processed by the network interface card.

4. The network interface card of claim 3, wherein said processing comprises taking over and performing at least one function of a physical layer from the host computer.

5. The network interface card of claim 3, wherein said processing comprises taking over and performing at least one function of a data link layer from the host computer.

6. The network interface card of claim 3, wherein said processing comprises taking over and performing at least one function of a network layer from the host computer.

7. The network interface card of claim 3, wherein said processing comprises taking over and performing at least one function of a transport layer from the host computer.

8. The network interface card of claim 3, wherein said ULP handler is adapted to communicate with a ULP driver of said host computer.

9. The network interface card of claim 1, wherein said TCP handler is adapted to communicate with a transport accelerator driver of said host computer.

10. The network interface card of claim 1, wherein said TCP handler is adapted to communicate with the network.

11. The network interface card of claim 1, wherein said link handler is adapted to communicate with a link driver of said host computer.

12. The network interface card of claim 1, wherein said link handler is adapted to communicate with a network.

13. The network interface card of claim 1, wherein said network interface card further comprises at least one of a transport accelerator, embedded accelerator, portable stack, embedded link driver and embedded applications.

14. The network interface card of claim 1, wherein said network interface card is capable of receiving commands from an enhanced stack belonging to said host, said enhanced stack being further capable of supporting session layer acceleration.

15. The network interface card of claim 14, wherein the commands between said enhanced stack and said network interface card are performed using acceleration primitives.

16. The network interface card of claim 15, wherein said network interface card handles only a subset said acceleration primitives sent to said network

interface card from a plurality of said acceleration primitives sent to a plurality of network interface card devices.

17. The network interface card of claim 15, wherein at least one of said acceleration primitives is used to establish a direct connection between ULP of said host and said ULP handler.

18. The network interface card of claim 17, wherein said direct connection enables at least one function associated with a TCP/IP layer to be processed on said network interface card.

19. The network interface card of claim 18, wherein said direct connection comprises enables transferring data to said network interface card from said host and transferring data from said network interface card to said host.

20. The network interface card of claim 19, wherein said transferring data to said network interface card includes at least one of a transfer in request, a success transfer in reply and a fail transfer in reply.

21. The network interface card of claim 19, wherein said transferring data from said network interface card includes at least one of a transfer out request, a success transfer out reply and a fail transfer out reply.

22. The network interface card of claim 15, wherein said acceleration primitives are enabled by the use of an application programming interface (API) for interfacing between said host and said network interface card, said API being further comprised of a plurality of message primitives.

23. The network interface card of claim 22, wherein at least one of said message primitives is a connection transfer in message primitive sent in order to transfer a connection for acceleration by the network interface card.

24. The network interface card of claim 23, wherein said connection transfer in message primitive further contains at least information required to start a new or continued processing of an existing connection.

25. The network interface card of claim 24, wherein said necessary information comprises at least one of a connection 4tuple, initial sequence number, unacknowledged sequence number, acknowledgement sequence number, current sent time stamp, current received timestamp and remote or local negotiated window scale values.

26. The network interface card of claim 23, wherein said connection transfer in message primitive includes a unique identifier for connection reference,

said unique identifier being further recognizable by said network interface card and by said host.

27. The network interface card of claim 22 wherein at least one of said message primitives is a connection transfer out message primitive capable of causing said network interface card to transfer a connection out of said network interface card.

28. The network interface card of claim 27, wherein said connection transfer out message primitive provides only a connection reference to said network interface card.

29. The network interface card of claim 22, wherein at least one of said message primitives is a connection disconnect message primitive, said message primitive being capable of causing said network interface card to gradually close a previously opened connection to said network interface card.

30. The network interface card of claim 29, wherein said connection disconnect message primitive provides only a connection reference to said network interface card.

31. The network interface card of claim 22, wherein at least one of said message primitives is a connection abort message primitive, said message primitive being capable of causing said network interface card to abort a previously opened connection to said network interface card.

32. The network interface card of claim 31, wherein said connection abort message primitive provides only a connection reference to said network interface card.

33. The network interface card of claim 22, wherein at least one of said message primitives is a connection shutdown of transmission message primitive, said message primitive being capable of causing said network interface card to gracefully close a write side of a connection of said NIC.

34. The network interface card of claim 33, wherein said connection shutdown of transmission message primitive provides only a connection reference to said network interface card.

35. The network interface card of claim 22, wherein at least one of said message primitives is a connection send message primitive, said message primitive being capable of causing a transmission of data over an established connection.

36. The network interface card of claim 35, wherein said connection send message primitive is associated with data related to a TCP/IP connection, said data is at least one of connection reference, list of buffers in the host memory and their length, said buffers containing data to be transferred over a connection.

37. The network interface card of claim 22, wherein at least one of said message primitives is a connection receive message primitive, said message primitive being capable of causing a reception of data over an established connection, the data being received by connection specific receive buffers in a host memory.

38. The network interface card of claim 37, wherein said connection receive message primitive is associated with data related to a TCP/IP connection, said data is at least one of connection reference, list of connection specific buffers in the host memory and their length.

39. The network interface card of claim 22, wherein at least one of said message primitives is a connection synchronization message primitive, said message primitive being capable of flushing existing message pipes between the host and said network interface card.

40. The network interface card of claim 39, wherein said connection synchronization message primitive further includes a connection reference.

41. The network interface card of claim 39, wherein a connection synchronization reply message primitive is sent in response to said connection synchronization message primitive.

42. The network interface card of claim 41, wherein said connection synchronization reply message primitive further includes a connection reference.

43. The network interface card of claim 22, wherein said network interface card is capable of sending a connection send notify message primitive, wherein said connection send notify message primitive notifies of a successful transfer of an amount of data related to an offloaded TCP connection.

44. The network interface card of claim 43, wherein data associated with said connection send notify message primitive includes at least one of a connection reference and amount of data successfully transferred over the connection.

45. The network interface card of claim 22, wherein at least one of said message primitives is a connection receive notify message primitive, said message

primitive being capable of notifying of the reception of additional data by said network interface card over a connection.

46. The network interface card of claim 45, wherein said additional data may be directed to one of an anonymous host buffer and a connection specific host buffer.

47. The network interface card of claim 46, wherein at least one of said message primitives is a asynchronous buffer message primitive, said message primitive being capable of posting said anonymous receive buffers to said network interface card.

48. The network interface card of claim 47, wherein said anonymous receive buffers are used for a received TCP data and a layer 2 data.

49. The network interface card of claim 47, wherein data associated with said asynchronous buffer message primitive includes a list of buffers in host memory and buffer lengths.

50. The network interface card of claim 45, wherein data associated with said connection receive notify message primitive includes connection reference, buffer identification and amount of data posted into the buffer.

51. The network interface card of claim 22, wherein said network interface card is capable of providing a notification from said network interface card to the host with an indication of a change in connection state.

52. The network interface card of claim 51, wherein the data associated with said notification includes connection reference, notification type and a connection state.

53. The network interface card of claim 52, wherein said notification type include connection established, connection disconnected, connection timed-out and connection gracefully closed.

54. A method of acceleration of a network operation, said method comprising:

- a) transferring a connection to a network interface card; and
- b) taking over from a host computer attached to the network and performing at least one network function at the network interface card.

55. The method of claim 54, wherein the network interface card is a layer 5 network interface card in a network implementing an open systems interconnection (OSI) protocol.

56. The method of claim 55, wherein data from the network is received and processed by the network interface card.

57. The method of claim 56, wherein said processing comprises taking over and performing at least one function of a physical layer from the host computer.

58. The method of claim 56, wherein said processing comprises taking over and performing at least one function of a data link layer from the host computer.

59. The method of claim 56, wherein said processing comprises taking over and performing at least one function of a network layer from the host computer.

60. The method of claim 56, wherein said processing comprises taking over and performing at least one function of a transport layer from the host computer.

61. A method for acceleration of a session layer network operation, said method comprising:

- a) sending a sequence of initialization commands from a ULP driver of a host to transport accelerator provider (TAP) of said host;
- b) sending a transfer message from said TAP to a TCP handler of a network interface card (NIC);
- c) sending from said NIC a synchronization command to a server over a network connecting said host computer and said server;
- d) receiving by said NIC a synchronization acknowledgement message over said network from said server;
- e) sending from said NIC an acknowledgment message to said server;
- f) sending a notification command to a ULP handler of said NIC;
- g) sending from said NIC a connection notification command to said TAP of said host; and,
- h) sending a connected information command to said ULP driver of said host.

62. The method of claim 61, wherein said method further comprises using an application programming interface (API) for interfacing between said host and said L5NIC, said API being further comprised of a plurality of message primitives.

63. The method of claim 62, wherein at least one of said message primitives is a connection transfer in message primitive sent in order to transfer a connection for acceleration by the network interface card.

64. The method of claim 63, wherein said connection transfer in message primitive further contains at least information required to start a new or continued processing of an existing connection.

65. The method of claim 64, wherein said necessary information comprises at least one of a connection 4tuple, initial sequence number, unacknowledged sequence number, acknowledgement sequence number, current sent time stamp, current received timestamp and remote or local negotiated window scale values.

66. The method of claim 63, wherein said connection transfer in message primitive includes a unique identifier for connection reference, said unique identifier being further recognizable by said network interface card and by said host.

67. The method of claim 62, wherein at least one of said message primitives is a connection transfer out message primitive capable of causing said network interface card to transfer a connection out of said network interface card.

68. The method of claim 67, wherein said connection transfer out message primitive provides only a connection reference to said network interface card.

69. The method of claim 62, wherein at least one of said message primitives is a connection disconnect message primitive, said message primitive being capable of causing said network interface card to gradually close a previously opened connection to said network interface card.

70. The method of claim 69, wherein said connection disconnect message primitive provides only a connection reference to said network interface card.

71. The method of claim 62, wherein at least one of said message primitives is a connection abort message primitive, said message primitive being capable of causing said network interface card to abort a previously opened connection to said network interface card.

72. The method of claim 71, wherein said connection abort message primitive provides only a connection reference to said network interface card.

73. The method of claim 62, wherein at least one of said message primitives is a connection shutdown of transmission message primitive, said message primitive being capable of causing said network interface card to gracefully close a write side of a connection of said NIC.

74. The method of claim 73, wherein said connection shutdown of transmission message primitive provides only a connection reference to said network interface card.

75. The method of claim 62, wherein at least one of said message primitives is a connection send message primitive, said message primitive being capable of causing a transmission of data over an established connection.

76. The method of claim 75, wherein said connection send message primitive is associated with data related to a TCP/IP connection, said data is at least one of connection reference, list of buffers in the host memory and their length, said buffers containing data to be transferred over a connection.

77. The method of claim 62, wherein at least one of said message primitives is a connection receive message primitive, said message primitive being capable of causing a reception of data over an established connection, the data being received by connection specific receive buffers in a host memory.

78. The method of claim 77, wherein said connection receive message primitive is associated with data related to a TCP/IP connection, said data is at least one of connection reference, list of connection specific buffers in the host memory and their length.

79. The method of claim 62, wherein at least one of said message primitives is a connection synchronization message primitive, said message primitive being capable of flushing existing message pipes between the host and said network interface card.

80. The method of claim 79, wherein said connection synchronization message primitive further includes a connection reference.

81. The method of claim 79, wherein a connection synchronization reply message primitive is sent in response to said connection synchronization message primitive.

82. The method of claim 81, wherein said connection synchronization reply message primitive further includes a connection reference.

83. The method of claim 62, wherein said network interface card is capable of sending a connection send notify message primitive, wherein said connection send notify message primitive notifies of a successful transfer of an amount of data related to an offloaded TCP connection.

84. The method of claim 83, wherein data associated with said connection send notify message primitive includes at least one of a connection reference and amount of data successfully transferred over the connection.

85. The method of claim 62, wherein at least one of said message primitives is a connection receive notify message primitive, said message primitive being capable of notifying of the reception of additional data by said network interface card over a connection.

86. The method of claim 85, wherein said additional data may be directed to one of an anonymous host buffer and a connection specific host buffer.

87. The method of claim 86, wherein at least one of said message primitives is a asynchronous buffer message primitive, said message primitive being capable of posting said anonymous receive buffers to said network interface card.

88. The method of claim 87, wherein said anonymous receive buffers are used for a received TCP data and a layer 2 data.

89. The method of claim 87, wherein data associated with said asynchronous buffer message primitive includes a list of buffers in host memory and buffer lengths.

90. The method of claim 85, wherein data associated with said connection receive notify message primitive includes connection reference, buffer identification and amount of data posted into the buffer.

91. The method of claim 62, wherein said network interface card is capable of providing a notification from said network interface card to the host with an indication of a change in connection state.

92. The method of claim 91, wherein the data associated with said notification includes connection reference, notification type and a connection state.

93. The method of claim 92, wherein said notification type include connection established, connection disconnected, connection timed-out and connection gracefully closed.

94. An application programming interface (API), said API facilitating interfacing between a host and a network interface card, said API comprising of a plurality of message primitives.

95. The API of claim 94, wherein at least one of said message primitives is a connection transfer in message primitive sent in order to transfer a connection for acceleration by the network interface card.

96. The API of claim 95, wherein said connection transfer in message primitive further contains at least information required to start a new or continued processing of an existing connection.

97. The API of claim 96, wherein said necessary information comprises at least one of a connection 4tuple, initial sequence number, unacknowledged sequence number, acknowledgement sequence number, current sent time stamp, current received timestamp and remote or local negotiated window scale values.

98. The API of claim 95, wherein said connection transfer in message primitive includes a unique identifier for connection reference, said unique identifier being further recognizable by said network interface card and by said host.

99. The API claim 94, wherein at least one of said message primitives is a connection transfer out message primitive capable of causing said network interface card to transfer a connection out of said network interface card.

100. The API of claim 99, wherein said connection transfer out message primitive provides only a connection reference to said network interface card.

101. The API of claim 94, wherein at least one of said message primitives is a connection disconnect message primitive, said message primitive being capable of causing said network interface card to gradually close a previously opened connection to said network interface card.

102. The API of claim 101, wherein said connection disconnect message primitive provides only a connection reference to said network interface card.

103. The API of claim 94, wherein at least one of said message primitives is a connection abort message primitive, said message primitive being capable of causing said network interface card to abort a previously opened connection to said network interface card.

104. The API of claim 103, wherein said connection abort message primitive provides only a connection reference to said network interface card.

105. The API of claim 94, wherein at least one of said message primitives is a connection shutdown of transmission message primitive, said message primitive

being capable of causing said network interface card to gracefully close a write side of a connection of said NIC.

106. The API of claim 105, wherein said connection shutdown of transmission message primitive provides only a connection reference to said network interface card.

107. The API of claim 94, wherein at least one of said message primitives is a connection send message primitive, said message primitive being capable of causing a transmission of data over an established connection.

108. The API of claim 107, wherein said connection send message primitive is associated with data related to a TCP/IP connection, said data is at least one of connection reference, list of buffers in the host memory and their length, said buffers containing data to be transferred over a connection.

109. The API of claim 94, wherein at least one of said message primitives is a connection receive message primitive, said message primitive being capable of causing a reception of data over an established connection, the data being received by connection specific receive buffers in a host memory.

110. The API of claim 109, wherein said connection receive message primitive is associated with data related to a TCP/IP connection, said data is at least one of connection reference, list of connection specific buffers in the host memory and their length.

111. The API of claim 94, wherein at least one of said message primitives is a connection synchronization message primitive, said message primitive being capable of flushing existing message pipes between the host and said network interface card.

112. The API of claim 111, wherein said connection synchronization message primitive further includes a connection reference.

113. The API of claim 111, wherein a connection synchronization reply message primitive is sent in response to said connection synchronization message primitive.

114. The API of claim 113, wherein said connection synchronization reply message primitive further includes a connection reference.

115. The API of claim 94, wherein said network interface card is capable of sending a connection send notify message primitive, wherein said connection send

notify message primitive notifies of a successful transfer of an amount of data related to an offloaded TCP connection.

116. The API of claim 115, wherein data associated with said connection send notify message primitive includes at least one of a connection reference and amount of data successfully transferred over the connection.

117. The API of claim 94, wherein at least one of said message primitives is a connection receive notify message primitive, said message primitive being capable of notifying of the reception of additional data by said network interface card over a connection.

118. The API of claim 117, wherein said additional data may be directed to one of an anonymous host buffer and a connection specific host buffer.

119. The API of claim 118, wherein at least one of said message primitives is a asynchronous buffer message primitive, said message primitive being capable of posting said anonymous receive buffers to said network interface card.

120. The API of claim 119, wherein said anonymous receive buffers are used for a received TCP data and a layer 2 data.

121. The API of claim 119, wherein data associated with said asynchronous buffer message primitive includes a list of buffers in host memory and buffer lengths.

122. The API of claim 117, wherein data associated with said connection receive notify message primitive includes connection reference, buffer identification and amount of data posted into the buffer.

123. The API of claim 94, wherein said network interface card is capable of providing a notification from said network interface card to the host with an indication of a change in connection state.

124. The API of claim 123, wherein the data associated with said notification includes connection reference, notification type and a connection state.

125. The API of claim 124, wherein said notification type include connection established, connection disconnected, connection timed-out and connection gracefully closed.

126. The API of claim 94, wherein a set parameters message primitive is capable of updating at least a global TCP/IP parameter of an TCP/IP stack of said NIC.

127. The API of claim 126, wherein associated data with said set parameters message primitive includes at least a list of parameters to be changed.

128. The API of claim 94, wherein a get parameters message primitive is sent to said L5NIC for the purpose of retrieving at least a global non-connection specific TCP/IP parameter from a TCP/IP stack of said NIC.

129. The API of claim 128, wherein associated data with said get parameters message primitive includes at least a list of parameters to be retrieved.

130. The API of claim 128, wherein in response to said get parameters message primitive a get parameters reply message primitive is sent by said NIC.

131. The API of claim 130, wherein associated data with said get parameters message primitive includes at least a list of reported values.

132. The API of claim 94, wherein a get statistics message primitive is sent to said L5NIC for the purpose of gathering statistics from a TCP/IP stack of said NIC on at least a link layer and TCP/IP.

133. The API of claim 132, wherein a get statistics reply message primitive is sent in response to said get statistics message primitive, and further providing a set of reported statistical values.

134. The method of claim 54, wherein the network interface card corresponds to a layer 5 of a seven layer OSI network architecture.

135. The method of claim 61, wherein the network interface card corresponds to a layer 5 of a seven layer OSI network architecture.